



HAL
open science

Comparison of the MATSuMoTo Library for Expensive Optimization on the Noiseless Black-Box Optimization Benchmarking Testbed

Dimo Brockhoff

► **To cite this version:**

Dimo Brockhoff. Comparison of the MATSuMoTo Library for Expensive Optimization on the Noiseless Black-Box Optimization Benchmarking Testbed. Congress on Evolutionary Computation (CEC 2015), May 2015, Sendai, Japan. hal-01157388

HAL Id: hal-01157388

<https://inria.hal.science/hal-01157388>

Submitted on 28 May 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Comparison of the MATSuMoTo Library for Expensive Optimization on the Noiseless Black-Box Optimization Benchmarking Testbed

Dimo Brockhoff

Inria Lille - Nord Europe, DOLPHIN team

Email: dimo.brockhoff@inria.fr

Abstract—Numerical black-box optimization problems occur frequently in engineering design, medical applications, finance, and many other areas of our society’s interest. Often, those problems have expensive-to-calculate objective functions for example if the solution evaluation is based on numerical simulations. Starting with the seminal paper of Jones et al. on Efficient Global Optimization (EGO), several algorithms tailored towards expensive numerical black-box problems have been proposed. The recent MATLAB toolbox MATSuMoTo (short for MATLAB Surrogate Model Toolbox) is the focus of this paper and is benchmarked within the Black-box Optimization Benchmarking framework BBOB. A comparison with other already previously benchmarked algorithms for expensive numerical black-box optimization with the default setting of MATSuMoTo highlights the strengths and weaknesses of MATSuMoTo’s cubic radial basis functions surrogate model in combination with a Latin Hypercube initial design in the range of 50 times dimension many function evaluations.

I. INTRODUCTION

Numerical black-box optimization problems, i.e., problems with continuous variables but without the availability of derivatives, have to be solved frequently in many businesses these days. The number of available function evaluations is thereby often restricted to about 10 to 1000 times the search space dimension DIM (*expensive setting*) as a typical evaluation of the objective function can take several minutes or even hours of (already parallelized) computation time.

A state-of-the-art approach to tackle expensive optimization problems is to build a so-called surrogate model of the objective function (based on already evaluated search points) and to use this easier-to-evaluate surrogate model to predict good candidate solutions. Several surrogate-assisted (or model-based) algorithms for expensive numerical black-box optimization problems exist which mainly differ in the type of the underlying model of the objective function (local vs. global,

Erratum: The statement “MATSuMoTo often scales linearly or quadratically with the problem dimension for solved problems while the expected running time for RANDOMSEARCH explodes exponentially.” in the original paper was wrong and has now been replaced by “For the solved problems, MATSuMoTo often scales linearly or quadratically relative to the best algorithm of BBOB 2009 while the expected running time for RANDOMSEARCH always explodes exponentially.”

This is an updated version of the original paper, published at IEEE Congress on Evolutionary Computation (CEC 2015). The original publication is available at <http://ieeexplore.ieee.org>.

quadratic model, Kriging, radial basis functions, etc.) and the way this model is used for optimization (for example wrt. the criterion for selecting candidate solutions). The criterion for choosing new candidate solutions (to be evaluated on the true, expensive objective function) is known under different names such as “figure of merit” or “infill criterion”.

The probably most known approach to expensive numerical black-box optimization is the *Efficient Global Optimization* algorithm (EGO) from the seminal paper by Jones et al. [1]. Here, multivariate *Gaussian processes* are used as surrogate models and the *expected improvement* as infill criterion.

The recently proposed SMAC-BBOB [2] is a similar approach to EGO and uses the same surrogate models and infill criterion as EGO. In comparison to EGO, however, SMAC-BBOB uses the specific “noise-free isotropic Matern kernel and no initial design” [2]. Furthermore, the optimization algorithms DIRECT and CMA-ES are used to optimize the expected improvement criterion instead of the branch-and-bound approach of the original EGO.

A local meta-model based version of the CMA-ES algorithm itself, denoted by Imm-CMA-ES, has been proposed by Kern et al. [3] and was later slightly improved [4]. As its original version CMA-ES, it samples in each iteration λ candidate solutions from a multivariate normal distribution which itself is updated based on the ranking of the candidate solutions’ objective function values. In the Imm-CMA-ES, a local quadratic surrogate model is build around each candidate solution to predict a ranking. Iteratively, only a small portion of the candidate solutions is then evaluated on the true objective function until the (updated) ranking does not change anymore.

Another variant of the original CMA-ES algorithm which uses surrogates is the so-called IPOPSaACM algorithm [5]. On top of a variant of CMA-ES that uses a ranking support vector machine as surrogate model, the IPOPSaACM proposes a heuristic that adapts both the number of function evaluations, within which the surrogate model is kept constant, and the surrogate’s model parameters itself.

Last, let us mention the algorithm NEWUOA by Powell [6], [7] which is not specifically designed for solving expensive optimization problems but also builds a (global) quadratic model of the objective function in each iteration and, thus, should be considered as a baseline in each comparison of

optimizers for expensive numerical black-box optimization problems. Instead of using quadratically many solutions to fit the quadratic surrogate model of NEWUOA, typically only linearly many solutions are used to define the surrogate. Minimizing the Frobenius norm of the second derivative matrix of the model changes in each iteration then makes up for the remaining freedom of the quadratic model.

In order to find out which of the many available optimization algorithms performs best on certain classes of functions, benchmarking in terms of numerical experiments is the compulsory path to assess performance of optimizers quantitatively and to understand weaknesses and strengths of each algorithm. To facilitate this tedious task, the Comparing Continuous Optimizers platform (COCO) has been developed and used to create the Black-box Optimization Benchmarking (BBOB) test suite [8]. It provides all necessary code for running the experiments on 24 well-known and -understood noiseless test functions, the collection of data, up to the automated postprocessing of them—including the generation of data profiles, scaling graphs, and tables. In the beginning of 2015, around 120 different algorithms have been benchmarked with the COCO/BBOB framework and the corresponding data sets are available online at <http://coco.gforge.inria.fr/>. However, the data collection is by far not exhaustive and in particular in the expensive setting (between $10 \cdot \text{DIM}$ and $1000 \cdot \text{DIM}$ function evaluations) does not cover many different algorithm classes.

Our Contributions: This paper will, therefore, benchmark a recently proposed approach to expensive optimization, the MATLAB Surrogate Model Toolbox (MATSuMoTo) [9], [10], and compare its performance on the BBOB noiseless testbed with other above mentioned algorithms. The MATSuMoTo library allows to choose from a variety of different initial designs, surrogate models, and criteria for the choice of new candidate solutions and a previous benchmarking of the available variants has established a default setting which is based on cubic radial basis functions (RBFs) as surrogate models [10]. Since cubic RBFs have not been used as surrogates in any algorithm previously compared on the expensive BBOB testbed, we turn our attention here to the comparison of this default setting of MATSuMoTo with other previously benchmarked algorithms for expensive optimization.

II. THE MATSuMoTo LIBRARY AND THE BENCHMARKED DEFAULT ALGORITHM SETTING

The MATLAB Surrogate Model Toolbox (MATSuMoTo) is an optimization toolbox for “computationally expensive, black-box, global optimization problems that may have continuous, mixed-integer, or pure integer variables” [9]. Various surrogate models, initial experimental design strategies and infill criteria are available. Also mixtures of surrogate models as employed and compared in [10] can be used. We here restrict ourselves to the continuous optimization part of the toolbox.

Several parameters have to be specified by the user when using MATSuMoTo, concretely the optimization problem, the maximum number of allowed expensive function evaluations,

the surrogate model type, the sampling strategy, the type of the initial experimental design, the number of points in the initial experimental design, and the number of points to be selected in each iteration for the expensive function evaluations. Optionally, specific points to be included in the initial experimental design can be specified. As the default setting of MATSuMoTo will be chosen by most new users of the toolbox as a starting point, we compare this default setting as a baseline version of MATSuMoTo within the BBOB framework. The default setting has been chosen based on a previous extensive comparison of all of MATSuMoTo’s components. The default setting of MATSuMoTo corresponds to using cubic radial basis functions as surrogate model, randomized sampling by local perturbation of the best point found so far together with additional points uniformly selected from the whole variable domain as sampling strategy, a Latin Hypercube sampling as initial experimental design with $2 \cdot (\text{DIM} + 1)$ samples, and one new sample per iteration. Only the stopping criterion was set differently than the default, namely to $50 \cdot \text{DIM}$ instead of $20 \cdot \text{DIM}$.

Slight modifications had to be done to the original MATSuMoTo code to be able to connect it to the BBOB framework. The main change is that no parallel evaluations are performed anymore via MATLAB’s Parallel Computing toolbox but instead the natural parallel evaluations of BBOB are used. As basis for our experiments, the online available MATSuMoTo version of April, 8, 2014 has been used. The source code for the BBOB experiments will be available via <http://coco.gforge.inria.fr/doku.php?id=cec-bbob-2015-results>.

III. EXPERIMENTAL SETTING

A. Compared Algorithms

Besides the above described default setting of MATSuMoTo, several algorithm data sets from the COCO/BBOB web page have been included in the comparison: NEWUOA [7], SMAC-BBOB [2], Imm-CMA-ES [4], IPOPsaACM [5] and pure random search [11]. The default MATSuMoTo optimizer has been run for $50 \cdot \text{DIM}$ function evaluations.

B. CPU Timing of the Default MATSuMoTo

In order to evaluate the CPU timing of the MATSuMoTo library, we have run the algorithm with default settings on the function f_8 with restarts for at least 30 seconds and until a maximum budget equal to $50 \cdot \text{DIM}$ is reached. The code was run on an Intel(R) Core(TM)2 Quad Q6600 CPU @ 2.40GHz with 1 processor and 4 cores. The time per function evaluation for dimensions 2, 3, 5, 10, and 20 equals 0.058, 0.12, 0.26, 0.89, and 2.9 seconds respectively.

C. BBOB-related Settings

Results from experiments according to [8] on the functions given in [12] are presented in Figures 1, 2 and 3 and in Tables I and II. The **expected running time (ERT)** therein depends on a given target function value, $f_t = f_{\text{opt}} + \Delta f$, and is computed over all relevant trials as the number of function evaluations executed during each trial while the best function value did not

reach f_t , summed over all trials and divided by the number of trials that actually reached f_t [8], [13]. **Statistical significance** is tested with the rank-sum test for a given target Δf_t using, for each trial, either the number of needed function evaluations to reach Δf_t (inverted and multiplied by -1), or, if the target was not reached, the best Δf -value achieved, measured only up to the smallest number of overall function evaluations for any unsuccessful trial under consideration.

IV. DISCUSSION OF THE RESULTS

When looking at the benchmarking results in Figs. 1, 2, and 3 and in Tables I and II, four main observations can be made:

Scaling with Dimension: Over all problems, MATSuMoTo ranges in performance between RANDOMSEARCH and the other surrogate model based algorithms when looking at the expensive scenario of $10 \cdot \text{DIM}$ function evaluations (Fig. 1). For the solved problems, MATSuMoTo often scales linearly or quadratically relative to the best algorithm of BBOB 2009 while the expected running time for RANDOMSEARCH always explodes exponentially.

Solvable Instances: In 20-D (5-D), MATSuMoTo does not solve nine (three) of the 24 noiseless BBOB functions to the precision of the BBOB-2009 reference algorithm after $10 \cdot \text{DIM}$ function evaluations. Surprisingly, the sphere function cannot be solved to a precision of 10^{-8} within $50 \cdot \text{DIM}$ evaluations—not even in 2-D. The only algorithms in the comparison for which this is also the case in 5-D and 20-D, are RANDOMSEARCH and SMAC-BBOB. Preliminary experiments with other MATSuMoTo settings show similar results but a thorough investigation remains future work.

Strong Performances of MATSuMoTo: The best relative performances of MATSuMoTo can be observed on the functions f_{15} (Rastrigin) and f_{21} (Gallagher’s 101 Peaks) in 5-D and on f_{21} and f_{22} (Gallagher’s 21 Peaks) in 20-D—providing the best performances among the compared algorithms. For the 20-D functions f_{21} and f_{22} and the largest expensive budget of $50 \cdot \text{DIM}$, MATSuMoTo is even outperforming the best algorithm of BBOB-2009. For smaller budgets than $50 \cdot \text{DIM}$ and in 5-D, also the results on f_2 , f_6 , and f_{16} are competitive and MATSuMoTo sometimes outperforms the best BBOB-2009 algorithm. None of the results is statistically significant.

Overall Comparison with Other Surrogate-Assisted Optimizers: When compared on the data profiles of Fig. 2 and 3, it appears that the default MATSuMoTo optimizer is always dominated by some other algorithm. Moreover, the combination of the three algorithms SMAC-BBOB (for very low budgets below $\approx 3 \cdot \text{DIM}$ function evaluations), NEWUOA (for medium budgets), and Imm-CMA-ES (for relatively large budgets of $\geq 30 \cdot \text{DIM}$ evaluations) build a good portfolio that constructs the upper envelope over all compared algorithms for almost all problem groups. Adding IPOPSaACM to the portfolio further improves performance slightly on the moderate function group for the most difficult targets.

V. CONCLUSIONS

The MATLAB Surrogate Model Toolbox (MATSuMoTo) has been taken out-of-the-box in its default setting and was compared with other model-building algorithms of the available BBOB algorithm data collection on the 24 noiseless test functions of the BBOB suite. It turns out that MATSuMoTo shows comparable results over most functions: though in dimension 20, nine functions cannot be solved to comparatively high precision, on the two Gallagher functions (f_{21} and f_{22}), the best BBOB-2009 algorithm is outperformed for the largest expensive budgets (all results not statistically significant).

Overall, MATSuMoTo is a practically interesting optimization toolbox due to its flexibility and availability in MATLAB. However, other available algorithms that outperform its default setting exist and it remains open to investigate more carefully the impact of the other options offered by the framework on the BBOB test suite, similar to the comparison in [10]. In particular, the influence of the initial design, the used surrogate model and the employed infill criterion in surrogate-assisted optimization algorithms should be investigated further.

ACKNOWLEDGMENT

This work was supported by the French National Research Agency under grant ANR-12-MONU-0009 (NumBBO), the JSPS funded project “Global Research on the Framework of Evolutionary Solution Search to Accelerate Innovation” and Inria’s Équipe Associée “s3-bbo”. Figures, Tables, and part of Sec. III stem from the open-source platform COCO [8].

REFERENCES

- [1] D. R. Jones, M. Schonlau, and W.-J. Welch, “Efficient Global Optimization of Expensive Black-Box Functions,” *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.
- [2] F. Hutter, H. Hoos, and K. Leyton-Brown, “An Evaluation of Sequential Model-Based Optimization for Expensive Blackbox Functions,” in *GECCO workshop on Black-Box Optimization Benchmarking (BBOB’2013)*. ACM Press, 2013, pp. 1209–1216.
- [3] S. Kern, N. Hansen, and P. Koumoutsakos, “Local Meta-models for Optimization Using Evolution Strategies,” in *Parallel Problem Solving from Nature (PPSN IX)*. Springer, 2006, pp. 939–948.
- [4] A. Auger, D. Brockhoff, and N. Hansen, “Benchmarking the Local Metamodel CMA-ES on the Noiseless BBOB2013 Test Bed,” in *GECCO (Companion) workshop on Black-Box Optimization Benchmarking (BBOB’2013)*. ACM, 2013, pp. 1225–1232.
- [5] I. Loshchilov, M. Schoenauer, and M. Sebag, “Black-box Optimization Benchmarking of IPOP-saACM-ES and BIPOP-saACM-ES on the BBOB-2012 Noiseless Testbed,” in *GECCO (Companion) workshop on Black-Box Optimization Benchmarking (BBOB’2009)*. New York, NY, USA: ACM, 2012, pp. 175–182.
- [6] M. J. D. Powell, “The NEWUOA Software for Unconstrained Optimization Without Derivatives,” Department of Applied Mathematics and Theoretical Physics, Cambridge University, Tech. Rep. DAMTP 2004/NA05, Nov. 2004.
- [7] R. Ros, “Benchmarking the NEWUOA on the BBOB-2009 Function Testbed,” in *GECCO (Companion) workshop on Black-Box Optimization Benchmarking (BBOB’2009)*. ACM, 2009, pp. 2421–2428.
- [8] N. Hansen, A. Auger, S. Finck, and R. Ros, “Real-parameter black-box optimization benchmarking 2012: Experimental setup,” INRIA, Tech. Rep., 2012. [Online]. Available: <http://coco.gforge.inria.fr/doku.php?id=cec-bbob-2015-downloads>
- [9] J. Müller, “MATSuMoTo: The MATLAB Surrogate Model Toolbox For Computationally Expensive Black-Box Global Optimization Problems,” *ArXiv e-prints*, Apr. 2014.

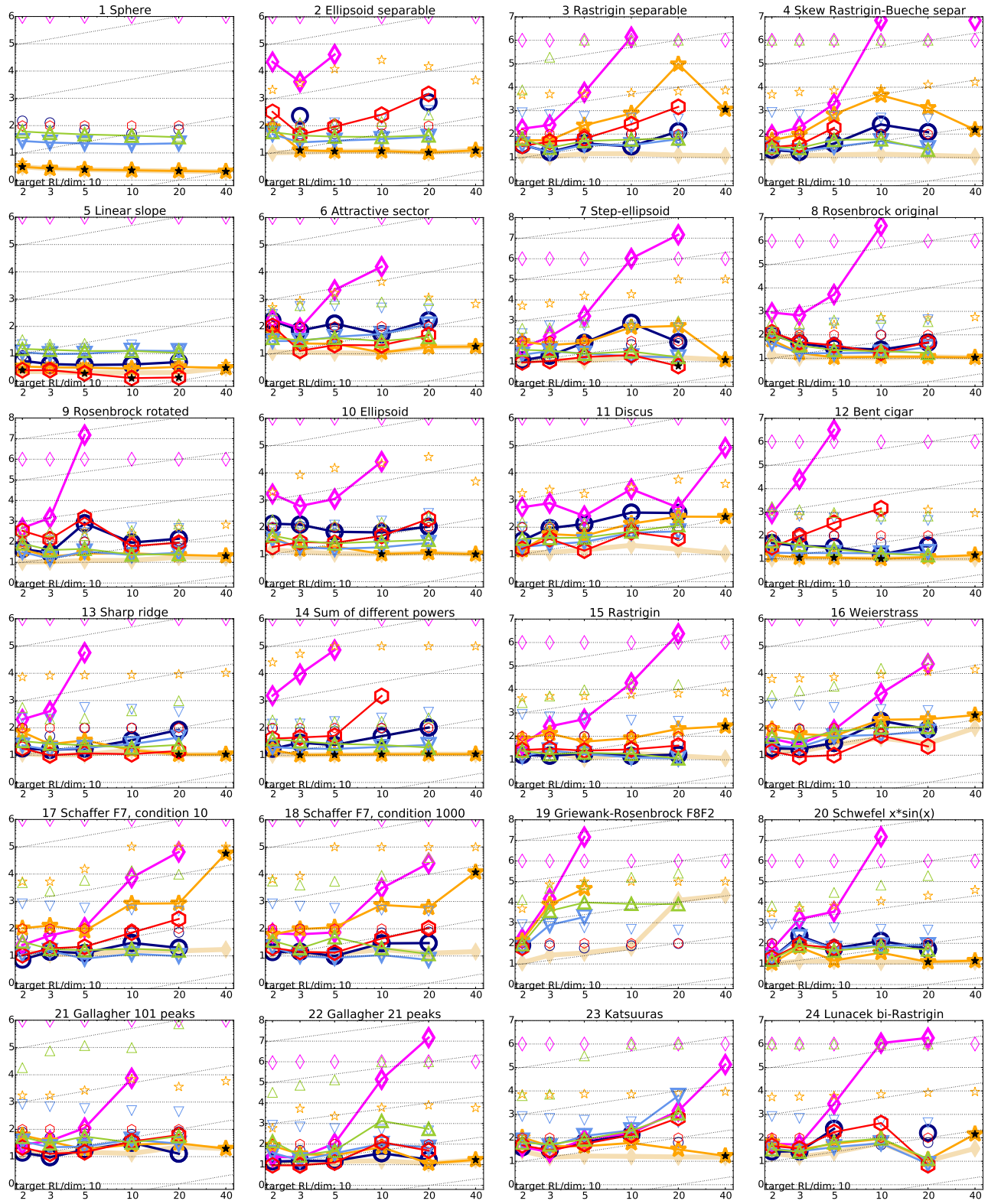


Fig. 1. Expected running time (ERT in number of f -evaluations as \log_{10} value) divided by dimension versus dimension. The target function value is chosen such that the bestGECCO2009 artificial algorithm just failed to achieve an ERT of $10 \times \text{DIM}$. Different symbols correspond to different algorithms given in the legend of f_1 and f_{24} . Light symbols give the maximum number of function evaluations from the longest trial divided by dimension. Black stars indicate a statistically better result compared to all other algorithms with $p < 0.01$ and Bonferroni correction number of dimensions (six). Legend: \circ :MATSuMoToDefault-50D, \diamond :RANDOMSEARCH, \star :NEUOJA, ∇ :lmm-CMA, \circ :SMAC, \triangle :IPOPsACM

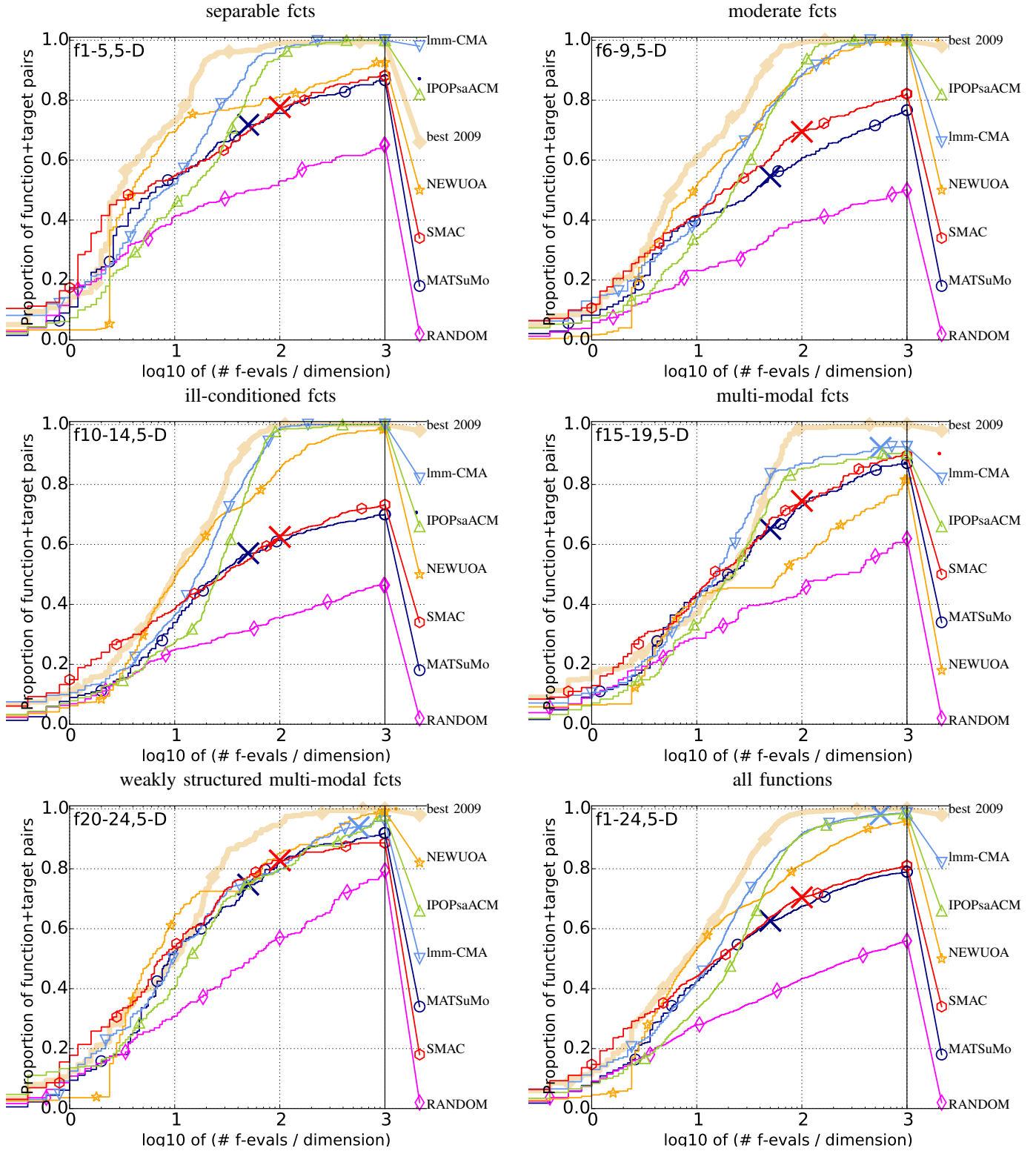


Fig. 2. Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) for all functions and subgroups in 5-D. The targets are chosen from $10^{[-8..2]}$ such that the bestGECCO2009 artificial algorithm just not reached them within a given budget of $k \times \text{DIM}$, with $k \in \{0.5, 1.2, 3, 10, 50\}$. The “best 2009” line corresponds to the best ERT observed during BBOB 2009 for each selected target.

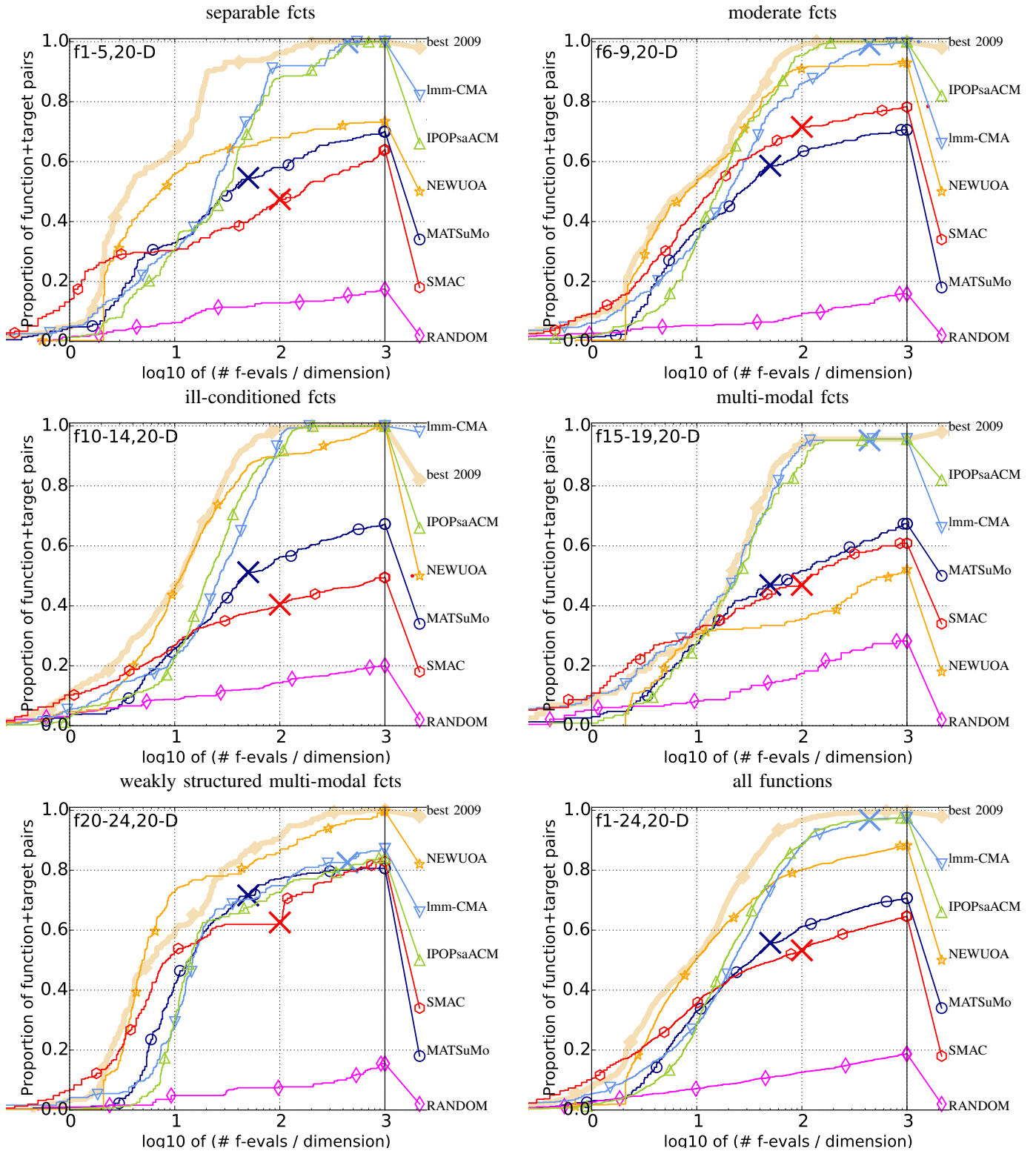


Fig. 3. Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) for all functions and subgroups in 20-D. The targets are chosen from $10^{[-8..2]}$ such that the bestGECCO2009 artificial algorithm just not reached them within a given budget of $k \times \text{DIM}$, with $k \in \{0.5, 1.2, 3, 10, 50\}$. The “best 2009” line corresponds to the best ERT observed during BBOB 2009 for each selected target.

#FES/D	0.5	1.2	3	10	50	#succ	#FES/D	0.5	1.2	3	10	50	#succ
F1	<i>6.3e+1.24</i>	<i>4.0e+1.42</i>	<i>1.0e-8.43</i>	<i>1.0e-8.43</i>	<i>1.0e-8.43</i>	15/15	F13	<i>1.6e+3.28</i>	<i>1.0e+3.64</i>	<i>6.3e+2.79</i>	<i>4.0e+1.211</i>	<i>2.5e+0.1724</i>	15/15
MATSuMo	2.5(0.5)	2.0(0.2)	∞	∞	∞ 1000	0/15	MATSuMo	2.6(0.5)	1.8(0.3)	2.6(2)	8.0(8)	∞ 1000	0/15
RANDOM	892(1140)	3.7e4(3e4)	∞	∞	∞ 2e7	0/15	RANDOM	358(119)	1.5e5(1e5)	∞	∞	∞ 2e7	0/15
NEUOEA	1.7(0)	1.0(0.0)	1.0(0.0)*4	1.0(6e-3)*4	1.0(0.0)*4	15/15	NEUOEA	1.7(0.3)	1.0(1)	1.0(2)	1.0(1)*3	1.5(2)	15/15
Imm-CMA	2.5(2)	2.5(0.9)	10(0.1)	10(0.3)	10(0.3)	15/15	Imm-CMA	1.6(0.5)	2.6(0.8)	3.3(0.9)	4.2(0.3)	1.1(0.2)	15/15
SMAC	0.80(0.3)*3	0.67(0.2)*4	∞	∞	∞ 2000	0/15	SMAC	0.81(0.5)*2	0.66(0.1)*3	0.84(0.1)*	1.4(0.2)	∞ 2000	0/15
IPOPSaACM	3.4(1)	2.8(0.5)	18(0.9)	18(0.7)	18(0.8)	15/15	IPOPSaACM	2.8(1)	2.6(0.2)	2.5(0.1)	2.3(3)	1.3(0.7)	15/15
#FES/D	0.5	1.2	3	10	50	#succ	#FES/D	0.5	1.2	3	10	50	#succ
F2	<i>4.0e+6.29</i>	<i>2.5e+6.42</i>	<i>1.0e+5.65</i>	<i>1.0e+4.207</i>	<i>1.0e-8.412</i>	15/15	F14	<i>2.5e+1.15</i>	<i>1.6e+1.42</i>	<i>1.0e+1.75</i>	<i>1.6e+0.219</i>	<i>6.3e-4.1106</i>	15/15
MATSuMo	0.79(0.7)	1.3(0.7)	9.3(10)	71(90)	∞ 1000	0/15	MATSuMo	5.7(2)	3.1(0.8)	2.9(1)	9.5(14)	∞ 1000	0/15
RANDOM	2.8(3)	4.4(4)	1.0e6(5e5)	∞	∞ 2e7	0/15	RANDOM	155(131)	1839(1455)	4.5e4(5e4)	∞	∞ 2e7	0/15
NEUOEA	1.4(0)	1(0)	1.0(4)*4	1(0.5)*4	303(190)	15/15	NEUOEA	4.3(1)	2.0(0.8)	1.5(0.6)*	1(0.3)*3	1(0.0,2)	15/15
Imm-CMA	0.53(0.5)	0.68(0.8)	7.2(1)	3.8(0.9)	14(2)	15/15	Imm-CMA	3.8(4)	2.9(2)	3.0(1)	2.2(0.5)	1.9(0.2)	15/15
SMAC	0.54(0.5)	0.70(0.4)	23(19)	143(131)	∞ 2000	0/15	SMAC	2.0(1)	3.3(8)	19(21)	∞	∞ 2000	0/15
IPOPSaACM	1.2(2)	1.4(1)	9.1(3)	4.3(0.8)	10(1)	15/15	IPOPSaACM	7.2(2)	4.2(0.6)	3.0(0.6)	1.8(0.2)	1.3(0.1)	15/15
#FES/D	0.5	1.2	3	10	50	#succ	#FES/D	0.5	1.2	3	10	50	#succ
F3	<i>6.3e+2.33</i>	<i>4.0e+2.44</i>	<i>1.6e+2.109</i>	<i>1.0e+2.255</i>	<i>2.5e+1.3277</i>	15/15	F15	<i>6.3e+2.15</i>	<i>4.0e+2.67</i>	<i>2.5e+2.292</i>	<i>1.6e+2.846</i>	<i>1.0e+2.1671</i>	15/15
MATSuMo	1.9(0.3)	2.5(0.8)	8.0(5)	10(16)	∞ 1000	0/15	MATSuMo	3.5(2)	1.7(0.9)	1.1(1)	2.0(2)	8.8(10)	1/15
RANDOM	9.1(4)	1397(1067)	∞	∞	∞ 2e7	0/15	RANDOM	35(45)	1316(1904)	1.6e5(2e5)	∞	∞ 2e7	0/15
NEUOEA	2.5(0.8)	2.4(2)	359(561)	7468(9609)	∞ 1e5	0/15	NEUOEA	4.3(2)	11(12)	14(27)	41(65)	1078(772)	1/15
Imm-CMA	1.0(0.4)	2.3(1)	4.8(0.9)	2.1(1.0)	14(15)	15/15	Imm-CMA	1.4(2)	1.3(1)	0.77(0.2)	0.72(0.3)	0.86(0.1)	15/15
SMAC	0.49(0.5)₁₂	2.1(0.9)	124(120)	114(157)	∞ 2000	0/15	SMAC	1.1(2)	2.9(0.9)	2.8(1)	∞	∞ 2000	0/15
IPOPSaACM	2.2(2)	3.3(0.7)	6.4(3)	5.1(0.8)	2.6(1)	5/5	IPOPSaACM	3.6(2)	2.1(0.6)	0.78(0.1)	0.96(0.6)	1.1(0.4)	15/15
#FES/D	0.5	1.2	3	10	50	#succ	#FES/D	0.5	1.2	3	10	50	#succ
F4	<i>6.3e+2.22</i>	<i>4.0e+2.91</i>	<i>2.5e+2.250</i>	<i>1.6e+2.332</i>	<i>6.3e+1.1927</i>	15/15	F16	<i>4.0e+1.26</i>	<i>2.5e+1.127</i>	<i>1.6e+1.540</i>	<i>1.6e+1.540</i>	<i>1.0e+1.1384</i>	15/15
MATSuMo	7.6(3)	5.1(6)	10(17)	∞	∞ 1000	0/15	MATSuMo	3.4(4)	4.9(5)	3.3(2)	3.3(3)	11(11)	1/15
RANDOM	254(335)	3.2e4(4e4)	∞	∞	∞ 2e7	0/15	RANDOM	3.9(5)	33(35)	830(1404)	830(1371)	6.5e4(7e4)	3/15
NEUOEA	38(45)	48(44)	104(65)	1704(1523)	∞ 2e5	0/15	NEUOEA	2.4(2)	6.5(9)	7.8(13)	7.8(8)	16(27)	15/15
Imm-CMA	1.3(1)	2.2(0.6)	2.0(0.8)	3.0(0.8)	2.0(2)	13/15	Imm-CMA	4.8(2)	8.1(2)	2.7(0.8)	2.7(1)	1.2(0.5)	15/15
SMAC	6.9(10)	102(78)	∞	∞	∞ 2000	0/15	SMAC	2.4(3)	1.5(2)	0.78(0.2)	0.78(0.5)	0.76(0.4)*	14/15
IPOPSaACM	5.8(2)	2.3(0.2)	1.7(0.4)	4.0(0.8)	2.7(2)	5/5	IPOPSaACM	4.5(6)	14(6)	4.0(1)	4.0(2)	1.8(2)	15/15
#FES/D	0.5	1.2	3	10	50	#succ	#FES/D	0.5	1.2	3	10	50	#succ
F5	<i>2.5e+2.19</i>	<i>1.6e+2.34</i>	<i>1.0e-8.41</i>	<i>1.0e-8.41</i>	<i>1.0e-8.41</i>	15/15	F17	<i>1.6e+1.11</i>	<i>1.0e+1.63</i>	<i>6.3e+0.305</i>	<i>4.0e+0.468</i>	<i>1.0e+0.1030</i>	15/15
MATSuMo	1.8(0.9)	1.3(0.1)	2.4(0.1)	2.4(1)	2.4(0.2)	15/15	MATSuMo	3.3(2)	2.2(2)	1.3(0.9)	32(23)	∞ 1000	0/15
RANDOM	8(0.9)	1833(2303)	∞	∞	∞ 2e7	0/15	RANDOM	16(2)	120(98)	4216(7687)	∞	∞ 2e7	0/15
NEUOEA	2.2(0)	1.2(7e-3)	1.6(0.4)	1.6(0.4)	1.6(0.4)	15/15	NEUOEA	5.3(2)	16(55)	55(74)	3447(4325)	∞ 2e6	0/15
Imm-CMA	1.8(0.5)	2.1(0.5)	6.1(1)	6.1(0.8)	6.1(0.8)	15/15	Imm-CMA	0.6(2)	1(0.6)	0.65(0.3)	0.79(0.2)	1.4(0.4)	14/15
SMAC	0.46(0.1)₁₂	0.33(0.1)*4	0.66(0.1)*3	0.66(0.1)*3	0.66(0.1)*3	15/15	SMAC	0.52(0.1)	0.92(2)	15(15)	61(62)	∞ 2000	0/15
IPOPSaACM	2.2(1)	2.3(0.7)	5.2(0.6)	5.2(0.5)	5.2(2)	15/15	IPOPSaACM	6.2(5)	2.5(0.9)	1.0(0.4)	1.0(0.3)	0.91(0.2)	15/15
#FES/D	0.5	1.2	3	10	50	#succ	#FES/D	0.5	1.2	3	10	50	#succ
F6	<i>2.5e+5.16</i>	<i>6.3e+4.43</i>	<i>1.6e+4.62</i>	<i>1.6e+2.353</i>	<i>1.6e+1.1078</i>	15/15	F18	<i>4.0e+1.116</i>	<i>2.5e+1.252</i>	<i>1.6e+1.430</i>	<i>1.0e+1.621</i>	<i>4.0e+0.1090</i>	15/15
MATSuMo	2.4(1)	1.4(0.3)	1.4(0.8)	9.5(13)	∞ 1000	0/15	MATSuMo	1.0(1)	2.4(2)	8.3(9)	∞	∞ 1000	0/15
RANDOM	263(608)	3.2e4(9e4)	1.7e5(4e5)	∞	∞ 2e7	0/15	RANDOM	23(70)	1995(613)	∞	∞	∞ 2e7	0/15
NEUOEA	2.4(1)	1.2(0.3)	1(0.3)	1(0.7)	1(0.2)	15/15	NEUOEA	20(56)	1013(1571)	1.2e4(6117)	∞ 2e6	∞ 2e6	0/15
Imm-CMA	1.6(1)	1.6(0.8)	2.0(0.9)	6.8(3)	9.0(4)	11/15	Imm-CMA	0.41(0.6) ₁	0.57(0.3)	0.77(0.3)	0.76(0.3)	0.85(0.2)	15/15
SMAC	1.6(1)	1.2(0.9)	1.6(0.7)	2.8(4)	∞ 2000	0/15	SMAC	0.31(0.2)₁₂	8.3(4)	20(25)	22(41)	∞ 2000	0/15
IPOPSaACM	3.5(2)	2.4(0.5)	2.1(0.6)	2.3(1.0)	1.6(0.4)	15/15	IPOPSaACM	1.1(0.7)	0.94(0.5)	0.99(0.4)	0.96(0.3)	0.99(0.3)	15/15
#FES/D	0.5	1.2	3	10	50	#succ	#FES/D	0.5	1.2	3	10	50	#succ
F7	<i>1.0e+3.11</i>	<i>4.0e+2.39</i>	<i>2.5e+2.74</i>	<i>6.3e+1.319</i>	<i>1.0e+1.1351</i>	15/15	F19	<i>1.6e-1.25e5</i>	<i>1.0e-1.34e5</i>	<i>6.3e-2.34e5</i>	<i>4.0e-2.34e5</i>	<i>2.5e-2.34e5</i>	3/15
MATSuMo	1.2(1)	2.2(0.4)	2.0(2)	5.6(6)	∞ 1000	0/15	MATSuMo	∞	∞	∞	∞	∞ 1000	0/15
RANDOM	3.2(11)	76(55)	1020(588)	9.2e5(9e5)	∞ 2e7	0/15	RANDOM	∞	∞	∞	∞	∞ 2e7	0/15
NEUOEA	3.6(1)	1.4(0.3)	1.0(0.6)	34(64)	∞ 5e5	0/15	NEUOEA	∞	∞	∞	∞	∞ 2e6	0/15
Imm-CMA	0.49(0.6)	1.2(1)	1.7(1)	1.0(0.3)	0.48(0.1)	15/15	Imm-CMA	∞	∞	∞	∞	∞ 8805	0/15
SMAC	0.58(0.5)	0.61(0.5)	0.49(0.3)*	0.39(0.2)*	0.57(0.3)	15/15	SMAC	∞	∞	∞	∞	∞ 2000	0/15
IPOPSaACM	1.9(2)	2.1(2)	1.8(0.8)	1.0(0.2)	1.0(0.5)	15/15	IPOPSaACM	0.65(0.9)	0.61(0.2)	0.93(0.7)	2.2(1)	3.5(2)	14/15
#FES/D	0.5	1.2	3	10	50	#succ	#FES/D	0.5	1.2	3	10	50	#succ
F8	<i>4.0e+4.19</i>	<i>2.5e+4.35</i>	<i>4.0e+3.67</i>	<i>2.5e+2.231</i>	<i>1.6e+1.1470</i>	15/15	F20	<i>1.6e+4.38</i>	<i>1.0e+4.42</i>	<i>2.5e+2.62</i>	<i>2.5e+0.250</i>	<i>1.6e+0.2536</i>	15/15
MATSuMo	4.5(1)	2.9(0.7)	3.3(2)	4.0(2)	∞ 1000	0/15	MATSuMo	2.2(0.8)	2.4(1)	4.6(1)	4.1(2)	∞ 1000	0/15
RANDOM	809(908)	4532(3126)	4.2e6(3e6)	∞	∞ 2e7	0/15	RANDOM	216(229)	1380(909)	∞	∞	∞ 2e7	0/15
NEUOEA	2.2(0)	1.3(0.4)	1(0.4)*3	1(0.6)	1(0.4)	15/15	NEUOEA	1.1(0)	1(0.3)	1(0.5)	1(0.5)*4	2.1(3)	15/15
Imm-CMA	2.2(2)	2.5(1)	3.3(1)	2.6(0.6)	1.3(0.5)	15/15	Imm-CMA	0.91(0.7)	1.4(1.0)	3.9(0.7)	6.6(5)	23(24)	2/15
SMAC	1.4(2)	1.5(1)	2.5(0.6)	4.1(5)	∞ 2000	0/15	SMAC	0.25(0.2)₁₂	0.46(0.2)₁₄	0.90(0.1)	∞	∞ 2000	0/15
IPOPSaACM	5.2(2)	3.4(0.7)	2.9(0.4)	1.5(0.6)	1.0(0.2)	15/15	IPOPSaACM	2.4(1)	2.7(1)	3.2(0.3)	4.0(2)	8.8(9)	15/15
#FES/D	0.5	1.2	3	10	50	#succ	#FES/D	0.5	1.2	3	10	50	#succ
F9	<i>1.0e+2.357</i>	<i>6.3e+1.560</i>	<i>4.0e+1.684</i>	<i>2.5e+1.756</i>	<i>1.0e+1.1716</i>	15/15	F21	<i>6.3e+1.36</i>	<i>4.0e+1.77</i>	<i>4.0e+1.77</i>	<i>1.6e+1.456</i>	<i>4.0e+0.1094</i>	15/15
MATSuMo	7.9(6)	∞	∞	∞	∞ 1000	0/15	MATSuMo	2.7(0.7)	1.9(1)	1.9(0.4)	0.56(0.5)	0.97(2)	9/15
RANDOM	∞	∞	∞	∞	∞ 2e7	0/15	RANDOM	943(230)	2.0e4(3e4)	2.0e4(3e4)	∞	∞ 2e7	0/15
NEUOEA	1.3(0.5)	1.1(0.2)	1(0.4)	1.0(0.3)	1.0(0.2)	15/15	NEUOEA	1.8(0.2)*2	1(0.2)*3	1(0.2)*3	1.4(3)	2.9(0.8)	15/15
Imm-CMA	1.6(0.4)	1.3(0.1)	1.1(0.2)	1.2(0.3)	2.1(0.5)	15/15	Imm-CMA	3.5(1)	2.5(0.9)	2.5(0.9)	1.3(1)	4.3(5)	11/15
SMAC	5.0(3)	26(13)											

- [10] J. Müller and C. A. Shoemaker, "Influence of Ensemble Surrogate Models and Sampling Strategy on the Solution Quality of Algorithms for Computationally Expensive Black-box Global Optimization Problems," *Journal of Global Optimization*, vol. 60, no. 2, pp. 123–144, 2014.
- [11] A. Auger and R. Ros, "Benchmarking the Pure Random Search on the BBOB-2009 Testbed," in *GECCO (Companion) workshop on Black-Box Optimization Benchmarking (BBOB'2009)*. ACM, 2009, pp. 2479–2484.
- [12] N. Hansen, S. Finck, R. Ros, and A. Auger, "Real-parameter black-box optimization benchmarking 2010: Noiseless functions definitions," INRIA, Tech. Rep. RR-6829, 2010, compiled December 10, 2014. [Online]. Available: <http://coco.gforge.inria.fr/doku.php?id=cec-bbob-2015-downloads>
- [13] K. Price, "Differential evolution vs. the functions of the second ICEO," in *Congress on Evolutionary Computation (CEC'1997)*, 1997, pp. 153–157.